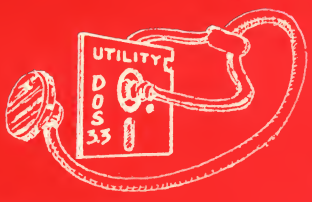
APPLE II SERIES

StripWare™

nibble®



DISK DOCTOR

Package is not returnable if plastic wrap is cut or removed.

NIBBLE MAGAZINE

If you own one of the Apple II series computers, your system is incomplete without Nibble magazine. Each issue is designed with both the beginner and expert Apple owner in mind. Every month, Nibble provides complete programs—ready to be typed in and run—accompanied by articles showing how to use them and how they work.

Nibble programs cover a wide variety of subjects including home/personal finance, business, programming utilities, education, and entertainment. Nibble also features insightful comparative product reviews, tutorials and tips on programming, techniques for using popular commercial software more efficiently, and information on the business aspect of professional and home computing.

Unique in the Apple magazine field, Nibble gives you something to *do* with your computer, not just something to read and discard. Nibble programs are available on disk or in Cauzin data strip form, at prices you can afford.

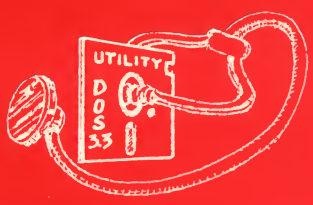
Now in its sixth year of publication, Nibble is published monthly by MicroSPARC Inc., producers of software, books, and Nibble Mac magazine for Macintosh enthusiasts.

Reprinted with permission of Nibble magazine.
Apple is a registered trademark of Apple Computer, Inc.
Softstrip and Stripware are trademarks of Cauzin Systems, Inc.
IBM is a registered trademark of International Business Machines, Inc.
Nibble is a registered trademark of MicroSPARC, Inc.

APPLE II SERIES

StripWare™

nibble®



DISK DOCTOR

Package is not returnable if plastic wrap is cut or removed.

Softstrip™

COMPUTER READABLE PRINT

Suggested Retail Price
$6.98

- Display and edit information directly from sectors of your disk
- Modify disk directories
- Read and write control characters
- Examine hidden file names
- Diagnose disk illnesses and cure them

Cauzin ©1983

CAUZIN SYSTEMS INC.
335 SOUTH MAIN STREET, WATERBURY, CT 06706

Welcome to the Cauzin Family

Welcome to the world of data strips. You now own a piece of a revolutionary technological advancement that bridges the gap between print and electronic media more quickly and easily than ever before.

Each data strip package contains selected highlights from among the best each publisher has to offer in articles, books, and software.

We at Cauzin Systems, Inc. hope you'll enjoy this software package, and believe you'll find data strips to be very practical and useful.

Your comments and ideas on this package or any additional applications you'd like to suggest, would be greatly appreciated. Please send your comments or remarks to:

CAUZIN PUBLISHING
835 South Main Street
Waterbury, Connecticut 06706

©1985



DISK DOCTOR

by Ben E. Colley

APPLE
UTILITIES

DISK DOCTOR does everything the name implies. It will help you put an end to whatever it is that is ailing your favorite disks.

The program contained in this package is typical of those you'll find each month in NIBBLE magazine.

They're paging Doctor Nibble. It's an emergency, seems a disk has gone bad. No problem . . . you step up to the operating system and read in DISK DOCTOR. We're sure the operation will be a success!

APPLE DISK DOCTOR



What do you call a program that helps diagnose disk illnesses and provides a remedy? We call it APPLE DISK DOCTOR, a disk utility that is easy to use and easy to enhance.

DISK DOCTOR is a member of the ZAP family of utilities. It allows you to accurately display, edit, and then write information **directly** to the sectors of the disk that you select. You can modify Disk Directories, read and write control characters, examine hidden names, and a whole host of other things.

DISK DOCTOR is written in Applesoft BASIC and POKEs in a short machine language program to interface with DOS 3.3. It requires at least 48K of memory and works on any Apple II computer. Be sure to put on CAPS LOCK if your Apple has it. Since the Doctor needs some important routines in DOS 3.3, it cannot be transferred to ProDOS. However, it can read from and write to a ProDOS disk.

Caution! Since DISK DOCTOR allows direct writing of disk sectors it should be tested and tried with a scratch/test diskette in order to become thoroughly familiar with it.

As you review the listing you will probably notice the lack of GOTO statements. DISK DOCTOR is a very structured, modular program using lots of GOSUB and FOR . . . NEXT logic. Let's go through the options and see what you can do with your disk.

Dump and Zap

Option #1 — Dump Sector. This section (**lines 1000-1310**) reads a sector and displays it on the screen, or the sector may be dumped to the printer if you have one (see **line 1120** for printer modifications if you do not have your printer in slot #1, or use other than the Apple parallel printer interface). The routine displays the sector in **hexadecimal** and **character format** in

40 column mode for screen displays, and in **80 column** mode for printed dumps.

Option #2 — Zap Sector. Lines 2000-2995 allow you to **change any byte** on the disk. The data is displayed one-half sector at a time; you may flip back and forth between the two pages by keying **CONTROL-P**. The Zap routine recognizes **five sub-commands**, namely:

Prescription

CONTROL-M — Move mode permits you to quickly **tab right, left, up, or down** via the **I, J, K, M keys**, positioning the cursor at the display cell you wish to edit. The cursor will wrap around the screen any time you tab beyond the display bounds.

CONTROL-E— Edit mode is where the "zapping" takes place. After positioning the cursor at the desired cell, **CONTROL-E** will move the cursor into the data cell. You may then key in the necessary hexadecimal digits directly over the current data. Each sector byte may be changed a nibble (half a byte) at a time. DISK DOCTOR calls a machine language routine (described later) to perform a nibble zap.

You may key in any valid hex digit (**0-F**) or use the right and left arrows to skip over data you do not wish to change. This mode also wraps around the screen when you backspace, forward space, or key past display bounds.

CONTROL-P — Display alternate half of sector.

CONTROL-W — Write the sector back to disk and return to MENU.

CONTROL-X — Return to MENU without writing sector.

As is probably apparent, you will spend most of your time in either MOVE or EDIT mode. At any time, you may switch to another mode, or flip display pages and remain in the current mode.

Recover Deleted Files

Option #3 — Recover File. Because we all make occasional mistakes and delete files that should not be deleted, **option 3** (**lines 3000-3550**) patches up the directory entry for the deleted file. Like most file recovery programs, the VTOC (Volume Table of Contents) bit map is NOT changed.

This means that when you have used DISK DOCTOR to recover a deleted file, the very first thing you should do is to **LOAD** the file and then **SAVE** it to the disk again. This sets up the VTOC to reserve the proper sectors to contain the file.

Also, please be cautioned that a deleted file must be recovered **before any other data or programs are written to the disk**. If you have written anything to the disk in the intervening period between deleting and recovering a file, the written data may have **overwritten** some of the sectors of the disk that were occupied by the deleted file.

DISK DOCTOR only looks for deleted files in this option; if you tell it to recover a file that has not been deleted, you will get a "NOT FOUND" message.



Remove DOS For More Space

Option #4 — Remove DOS. If you are like most Apple users I know, you only boot your system with one or two disks. The rest all have a copy of DOS on them that is simply taking up valuable disk space. **Option 4** (**lines 4000-4210**) does two things: **1**) make available an **additional 32 sectors** (track **1** and **2**), and **2**) places a **"mini-boot"** routine on track 0, sector 0. (DOS will not use track 0, and that space cannot be recovered.) If you try to boot a disk that has had the DOS removed in this manner you will get a message indicating that there is no DOS on the disk.

Slot/Drive Control

Option #5 — Change Slot and/or Drive. The program is set up to use both drives in a two drive system. If you change drives and then end the program, the slot and drive will revert to the values they had when you ran DISK DOCTOR. This is because DISK DOCTOR only changes the **IOB**, and not the DOS internal values for the slot and drive (**lines 5000-5090**).

How It Works

When you first **RUN** DISK DOCTOR, initialization is performed by **Lines 10000-10280**. From that time on, **lines 20-50** control the program's flow. First the **MENU** is presented via a **GOSUB** to **line 500**, and when a valid selection is keyed, control is **RETURN**ed at **line 600**. Then, based on the selection (**SEL**), one of the six subroutines outlined above is called to perform the requested function.

Notice the "**endless loop**" at **lines 20-50**. I use this technique heavily in all of my programs, and DISK DOCTOR is no exception. This little trick eliminates the need for countless GOTO statements! (ED. NOTE: This approach speeds up program execution since GOTO a statement forces the program to search for the line number starting at the beginning — lowest line number — of the program each time a GOTO is executed).

Another trick to eliminate GOTO statements is simulating the **ELSE statement** (which Applesoft does not have.) The following example shows how to handle "ELSE" conditions without using GOTO's.

```
If cond-A THEN statement
IF NOT cond-A THEN statement
```

Using the "simulated ELSE" eliminates the line number search (referred to above.) It should be apparent that restricting the use of GOTO statements will help speed program execution. Enough rambling; lets look at the machine language subroutines used by DISK DOCTOR.

Machine Language

Four machine language routines are used during execution of DISK DOCTOR. The first one is used only during initialization to obtain the address of DOS's **IOB** (Input/Output Block). The second provides the RWTS interface. The third provides the hex dump interface. The fourth routine is the nibble zap used by option two.

The routines reside in **page three of memory** at locations **770-833** (**$302-$341**), and are **POKE**d into memory by **line 10020**. Entry points to the three frequently used routines are at **780($30C)**, **800($320)**, **810($32A)**, and are referred to by DISK DOCTOR as **RWTS, PRBYTE,** and **ZAP**, respectively.

RWTS (Read Write Track Sector)

DISK DOCTOR interrogates the **IOB** (DOS Input/Output Block that controls the disk) error field after each call to the **RWTS** interface. RWTS (in DOS) returns a "clear" carry bit in the status register **(P)** but does not clear the **IOB** error field if the call is successful. The interface insures that the status register will be **zero (0)** if there were no errors. If there was an error, the RWTS (in DOS) sets this field and the carry bit. The interface simply exits to Applesoft with an error message (Line 11000) when a DOS error occurs.

PRBYTE Routine

The hex dump routine that is used to display the sector is really quite simple. The data byte to be displayed is POKEd into memory location 8. PRBYTE is then called (e.g. line 1160), and the interface routine calls the "real" (Apple Monitor) PRBYTE routine as described on Page 61 of the Apple Reference manual, using the byte at location 8.



Zap Disk

Because the machine language zap routine changes "nibbles" (half bytes) in the sector, it must know whether to change the upper nibble or the lower one. This information is passed as part of the new nibble data. If the new nibble value has the high bit set (i.e., value is greater than 127), then the left (high) nibble is to be "zapped", otherwise, the right (low) nibble is "zapped". Program statements 2500-2535 prepare for and make the CALL to zap the sector byte.

Information passed to the zap routine includes the displacement of the sector byte to change, the new nibble value, and the left or right nibble flag as just mentioned.

To pass the required parameters, DISK DOCTOR used the simplest method known, POKE-ing the variable data into known locations. Examination of the initialization at lines 10040 through 10220 shows the parameter fields are named such that they may be referenced symbolically by the program. This makes the program easier to read to debug and later on, to enhance!

Summary

While there are many "disk zap" routines available, we think DISK DOCTOR will be a good addition to your library. Even if you already have a disk utility, we're sure you can use some of the techniques implemented in DISK DOCTOR.

CAUZIN'S CORNER . . .

and now for something slightly different

The Doctor is easy to enhance. To add another option, follow these steps.

1) add a line to the menu. For example:

```
575 HTAB 12: PRINT "7) ONE MORE THING": PRINT
```

2) change the value towards the end of line 580, as shown below, from a "7" to an "8". This allows one more option value.

```
580 FOR XX = 1 TO 2 STEP 0: VTAB 22: INPUT "ENTER
SELECTION: "; S$: IF S$ > "0" AND S$ < "7"
THEN XX = 2
```

3) change line 40 to allow your option. The present line reads:

```
40 ON SEL GOSUB 1000,2000,3000,4000,5000,6000
```

By adding ",7000" to the end, you can add a new option that begins at line 7000. There is plenty of room in the 7000-9999 line number range. Remember the options are called using a GOSUB, so be sure to RETURN upon completion.

To examine the Doctor, type LOAD DISK DOCTOR and press the RETURN key. When the disk drive stops whirring, enter LIST to see the program. Typing LIST 500 will show you just the line you asked for, line 500 in this case.

What additional option might you wish to add? One feature is a routine to read sector X and write it to sector Y. However, don't try such sophisticated swapping until you are comfortable with the principles behind DOS.

If you are not familiar with the organization of information on disks, here are some good references:

1) the Apple DOS or ProDOS manual that came with your System Master.

2) *Beneath Apple DOS* and *Beneath Apple ProDOS*, by Don Wirth and Pieter Lechner, published by Quality Software.

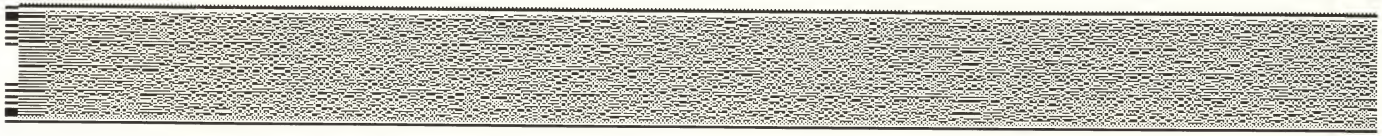
The Beneath series is packed full of information. You really should be comfortable with disk storage concepts before you use DISK DOCTOR; otherwise you might really "ZAP" a disk!

Reading DISK DOCTOR

On the next two pages are the data strips containing the DISK DOCTOR program. If you need additional help reading a data strip, refer to your reader instruction booklet. Your Cauzin Communications program also contains handy help screens to assist you.

After you've read in all the strip, run the program. The program is menu-driven and *only works with DOS 3.3*

To exit at any time simply press CONTROL-RESET.



Softstrip
COMPUTER READABLE PRINT

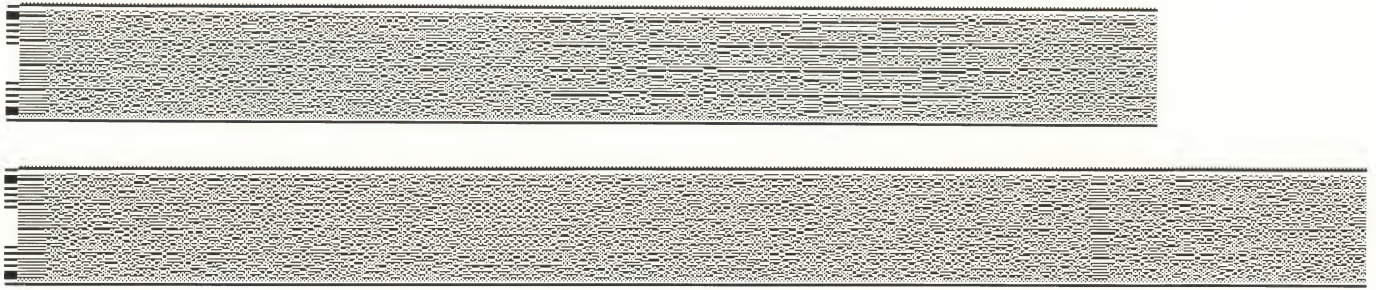
DISK DOCTOR

by Ben E. Colley

Nibble magazine Vol. III

Copyright 1983. All rights reserved.

Softstrip
COMPUTER READABLE PRINT



| 3 | 2

DISK DOCTOR

by Ben E. Colley

Nibble magazine Vol. III

Copyright 1983. All rights reserved.

DISK DOCTOR

```
10  GOSUB 10000
11 :
12 : REM      DISK DOCTOR
13 :
14 : REM       Ben Colley
15 : REM          1/82
16 :
17 : REM  (C) Micro-Sparc 1981
18 :
20  FOR I = 1 TO 2 STEP 0
30  GOSUB 500
40  ON SEL GOSUB 1000,2000,3000,4
    000,5000,6000
50  NEXT
60  TEXT : HOME : END
500  HOME
505  VTAB 5: HTAB 15: INVERSE : PRINT
    " OPTIONS ": NORMAL
510  VTAB 8
520  HTAB 12: PRINT "1) DUMP SECT
    OR": PRINT
530  HTAB 12: PRINT "2) ZAP SECTO
    R": PRINT
540  HTAB 12: PRINT "3) RECOVER F
    ILE": PRINT
550  HTAB 12: PRINT "4) REMOVE DO
    S": PRINT
560  HTAB 12: PRINT "5) SET DRIVE
     PARMS": PRINT
570  HTAB 12: PRINT "6) END": PRINT

580  FOR XX = 1 TO 2 STEP 0: VTAB
    22: INPUT "ENTER SELECTION:"
    ;S$: IF S$ > "0" AND S$ < "7
    " THEN XX = 2
590  NEXT
600 SEL =  VAL (S$): RETURN
1000  HOME
1010  FOR XX = 1 TO 2 STEP 0: VTAB
    7: HTAB 5: INPUT "ENTER TRAC
    K:";S$:TRK =  VAL (S$): IF T
    RK >  - 1 AND TRK < 35 AND S
    $ <  > "" THEN XX = 2
1020  IF  NOT (S$ >  = "0" AND S$
     <  = "9") THEN XX = 1
1030  NEXT
1040  FOR XX = 1 TO 2 STEP 0: VTAB
    9: HTAB 5: INPUT "ENTER SECT
    OR:";S$:SEC =  VAL (S$): IF
    SEC >  - 1 AND SEC < 16 AND
    S$ <  > "" THEN XX = 2
1050  IF  NOT (S$ >  = "0" AND S$
     <  = "9") THEN XX = 1
1060  NEXT
1070  FOR XX = 1 TO 2 STEP 0: VTAB
    11: HTAB 5: INPUT "LIST ON P
    RINTER? (Y/N):";S$: IF S$ =
    "Y" OR S$ = "N" THEN XX = 2
1080  NEXT
1090  POKE ITRK,TRK: POKE ISEC,
    C
1100  POKE ICMD,RD: CALL RWTS
1110  IF  PEEK (IERR) <  > 0 THEN
     GOSUB 11000: RETURN
1120  IF S$ = "Y" THEN INC = 16: PRINT
     CHR$ (4);"PR#1": PRINT  CHR$
    (9);"80N";: PRINT "TRACK=";T
    RK;"   SECTOR=";SEC: PRINT
1130  IF S$ = "N" THEN INC = 8: HOME
    : PRINT "        TRACK=";TRK
    ;"    SECTOR=";SEC: PRINT
1140  FOR XX = BUF TO BUF + 256 -
    INC STEP INC
1150  IF XX = BUF + 128 AND S$ =
    "N" THEN  VTAB 22: INPUT "PR
    ESS RETURN TO CONTINUE:";X$:
     VTAB 5: CALL ESCF
1160  POKE BYTE,XX - BUF: CALL PR
    BYTE
1170  PRINT ":  ";
1180  FOR XY = 0 TO INC - 1
1190  POKE BYTE, PEEK (XX + XY): CALL
    PRBYTE: REM   PRINT 1 BUFFER
     BYTE
1200  IF  INT (XY / 2) * 2 <  > X
    Y THEN  PRINT " ";
1210  NEXT
1220  PRINT "   :";
1230  FOR XY = 0 TO INC - 1
1240 XZ =  PEEK (XX + XY): IF XZ >
    127 THEN XZ = XZ - 128
1250  IF XZ < 32 THEN XZ = 46
1260  PRINT  CHR$ (XZ);
1270  NEXT
1280  PRINT ":": NEXT
1290  IF S$ = "N" THEN  VTAB 22: INPUT
    "PRESS RETURN TO CONTINUE:";
    X$
1300  IF S$ = "Y" THEN  PRINT : PRINT
    : PRINT  CHR$ (4);"PR#0"
1310  RETURN
2000  HOME
2005  FOR XX = 1 TO 2 STEP 0: VTAB
    7: HTAB 5: INPUT "ENTER TRAC
    K:";S$:TRK =  VAL (S$): IF T
    RK >  - 1 AND TRK < 35 AND S
    $ <  > "" THEN XX = 2
2010  IF  NOT (S$ >  = "0" AND S$
     <  = "9") THEN XX = 1
2015  NEXT
2020  FOR XX = 1 TO 2 STEP 0: VTAB
    9: HTAB 5: INPUT "ENTER SECT
    OR:";S$:SEC =  VAL (S$): IF
    SEC >  - 1 AND SEC < 16 AND
    S$ <  > "" THEN XX = 2
2025  IF  NOT (S$ >  = "0" AND S$
     <  = "9") THEN XX = 1
2030  NEXT
2035  POKE ITRK,TRK: POKE ISEC,SE
    C
```

```
2040  POKE ICMD,RD: CALL RWTS
2045  IF  PEEK (IERR) <  > 0 THEN
      GOSUB 11000: RETURN
2050  HOME : PRINT "         TRACK
      =";TRK;"    SECTOR=";SEC: PRINT

2055  VTAB 22: HTAB 1: PRINT " >A
      GE FLIP   >OVE   >DIT   >RIT
      E   >EXIT";
2060  INVERSE : HTAB 1: VTAB 22
2065  PRINT "P";: HTAB 13:: PRINT
      "M";: HTAB 20: PRINT "E";: HTAB
      27: PRINT "W";: HTAB 35: PRINT
      "X";
2070  NORMAL
2075 B1 = 0:B2 = 128:T =  - 128: GOSUB
      2800
2080 VK = 1:SEL = 13: REM MODE=MO
      VE
2085  FOR XX = 1 TO 2 STEP 0
2090  IF VK = 0 THEN  GOSUB 2975
2095  IF SEL = 5 THEN  GOSUB 2400

2100  IF SEL = 13 THEN  GOSUB 270
      0
2105  IF SEL = 16 THEN B1 = B1 +
      T:B2 = B2 + T: GOSUB 2800
2110  IF SEL = 23 THEN XX = 2
2115  IF SEL = 24 THEN XX = 2
2120  NEXT
2125  IF SEL = 23 THEN  POKE ICMD
      ,WR: CALL RWTS: IF  PEEK (IE
      RR) <  > 0 THEN  GOSUB 11000
2130  RETURN
2393  REM
2395  REM  EDIT MODE
2398  REM
2400  VTAB 3: HTAB 34: INVERSE : PRINT
      "EDIT";: NORMAL
2402  VTAB 24: HTAB 6: PRINT "<0-
      F>-HEX DIGIT   OR  <-, ->";
2405  FOR XY = 1 TO 2 STEP 0
2410 HT = HT + 1: REM  MOVE TO NE
      XT NIBBLE
2415  IF HT > 24 THEN HT = 6:VT =
      VT + 1: IF VT > 20 THEN VT =
      5
2420  IF  INT (HT / 5) * 5 = HT THEN
      HT = HT + 1: REM  CHECK FOR
      COL 10,15,20
2425  GOSUB 2975:VK = 0
2430  IF (S$ >  = "0" AND S$ <  =
      "9") OR (S$ >  = "A" AND S$ <
       = "F") THEN  GOSUB 2500:VK =
      1
2435  IF SEL = 13 THEN VK = 1:XY =
      2: REM  SWITCH TO MOVE
2440  IF SEL = 16 THEN B1 = B1 +
      T:B2 = B2 + T: GOSUB 2800: REM
       PAGE FLIP
2445  IF SEL = 23 THEN VK = 1:XY =
      2: REM  EXIT
2450  IF SEL = 24 THEN VK = 1:XY =
      2: REM  WRITE
2455  IF SEL = 21 THEN VK = 1: REM
       RIGHT ARROW
2460  IF SEL = 8 THEN VK = 1:HT =
      HT - 2: IF HT < 5 THEN HT =
      24:VT = VT - 1: IF VT < 5 THEN
      VT = 20: REM   LEFT ARROW
2465  IF SEL = 8 AND ( INT ((HT +
      1) / 5) * 5 = HT + 1) THEN H
      T = HT - 1: REM  CHECK COL 1
      0,15,20
2470  IF VK = 0 THEN HT = HT - 1:
       REM  NO VALID KEY
2475  NEXT
2480  REM  SET HT TO NEXT CELL BE
      FORE LEAVING
2485 HT =  INT ((HT + 4) / 5) * 5
      : IF HT > 20 THEN HT = 5:VT =
      VT + 1: IF VT > 20 THEN VT =
      5
2488  VTAB 24: HTAB 1: PRINT  SPC(
      35);
2490  RETURN
2495  REM  PREPARE FOR AND CALL Z
      AP
2496  REM   CALC BUFFER DISPLACEM
      ENT
2500 X1 = (VT - 5) * 8 + ZTBL(HT -
      6) - 30
2505  IF  PEEK (1536) = 184 THEN
      X1 = X1 + 128: REM  NIBBLE I
      N 2ND HALF OF SECTOR
2510  POKE BDISP,X1
2515 BIT = 0
2520  IF ZTBL(HT - 6) = ZTBL(HT -
      5) THEN BIT = 128: REM  DET
      IF LEFT NIBBLE
2525  IF S$ <  = "9" THEN  POKE B
      YTE, VAL (S$) + BIT
2530  IF S$ >  = "A" THEN  POKE B
      YTE, ASC (S$) - 55 + BIT
2535  CALL ZAP: REM  ZAP THE NIBB
      LE
2540  REM  NOW UPDATE DISPLAY
2544  VTAB VT
2545  HTAB HT: PRINT S$;: REM  PR
      INT NIBBLE
2550 X1 =  PEEK (BUF + X1): REM
      GET NEW VALUE
2555  VTAB VT: HTAB ZTBL(HT - 6):
       REM POS FOR CHAR
2560  IF X1 > 127 THEN X1 = X1 -
      128
2565  IF X1 < 32 THEN X1 = 46
2570  PRINT  CHR$ (X1);: REM  PRI
      NT CHARACTER
2575  RETURN
2685  REM
```

```
2690  REM  MOVE MODE
2695  REM
2700  VTAB 3: HTAB 34: INVERSE : PRINT
     "MOVE";: NORMAL
2705  VTAB 24: HTAB 3: PRINT "<I>
     -UP <M>-DOWN <J>-LEFT <K>-RI
     GHT";
2710  FOR XY = 1 TO 2 STEP 0
2715  GOSUB 2975:VK = 0
2720  IF S$ = "K" THEN HT = HT +
     5: IF HT > 20 THEN HT = 5:VT
      = VT + 1: IF VT > 20 THEN V
     T = 5
2725  IF S$ = "J" THEN HT = HT -
     5: IF HT < 5 THEN HT = 20:VT
      = VT - 1: IF VT < 5 THEN VT
      = 20
2730  IF S$ = "I" THEN VT = VT -
     1: IF VT < 5 THEN VT = 20
2735  IF S$ = "M" THEN VT = VT +
     1: IF VT > 20 THEN VT = 5
2740  IF SEL = 5 THEN VK = 1:XY =
     2
2745  IF SEL = 16 THEN B1 = B1 +
     T:B2 = B2 + T: GOSUB 2800
2750  IF SEL = 23 THEN VK = 1:XY =
     2
2755  IF SEL = 24 THEN VK = 1:XY =
     2
2760  NEXT
2765  VTAB 24: PRINT  SPC( 38);
2770  RETURN
2800  POKE 35,21: VTAB 5: HTAB 1:
      CALL ESCF: POKE 35,24: REM
       CLEAR MIDDLE OF SCREEN
2805  FOR X1 = BUF + B1 TO BUF +
     B2 - 8 STEP 8
2810  POKE BYTE,X1 - BUF: CALL PR
     BYTE
2815  PRINT ":  ";
2820  FOR X2 = 0 TO 7
2825  POKE BYTE, PEEK (X1 + X2): CALL
     PRBYTE
2830  IF  INT (X2 / 2) * 2 <  > X
     2 THEN  PRINT " ";
2835  NEXT
2840  PRINT "   :";
2845  FOR X2 = 0 TO 7
2850 X3 =  PEEK (X1 + X2)
2855  IF X3 > 127 THEN X3 = X3 -
     128
2860  IF X3 < 32 THEN X3 = 46
2865  PRINT  CHR$ (X3);
2870  NEXT
2875  PRINT ":"
2880  NEXT
2885 T =  - T
2890 VK = 0:SEL = 0:VT = 5:HT = 5
     : REM   INIT CURSOR POSITION

2895  RETURN
2975  VTAB VT: HTAB HT: REM POSIT
     ION CURSOR
2980  GET S$: PRINT  CHR$ (0): REM
      PRINT NON-DISPLAY CHAR
2985  IF S$ <  > "" THEN SEL =  ASC
     (S$)
2990  IF S$ =  CHR$ (0) THEN SEL =
     0
2995  RETURN
3000  FOR XX = 1 TO 2 STEP 0: HOME
     : VTAB 7
3010  HTAB 3: INPUT "ENTER FILE N
     AME:";S$
3020  IF  LEN (S$) <  > 0 THEN XX
      = 2
3030  NEXT
3040  POKE ITRK,17
3050 XU = 0
3060  FOR XX = 15 TO 1 STEP  - 1
3070  POKE ISEC,XX
3080  POKE ICMD,RD: CALL RWTS
3090  IF  PEEK (IERR) <  > 0 THEN
      GOSUB 11000: RETURN
3100  FOR XY = 11 TO 221 STEP 35
3110  IF  PEEK (BUF + XY) = 255 THEN
      GOSUB 3500
3120  IF XU = 1 THEN  POKE (BUF +
     XY), PEEK (BUF + XY + 32): POKE
     (BUF + XY + 32),160: POKE IC
     MD,WR: CALL RWTS:XY = 221:XX
      = 1: IF  PEEK (IERR) <  > 0
      THEN  GOSUB 11000: RETURN
3130  IF  PEEK (BUF + XY) +  PEEK
     (BUF + XY + 1) = 0 THEN XY =
     221:XX = 1
3140  NEXT : NEXT
3150  VTAB 18: HTAB 5
3160  IF XU = 1 THEN  PRINT "FILE
      RECOVERED IN VTOC": VTAB 20
     : HTAB 5: PRINT "REMEMBER TO
      LOAD AND SAVE IT"
3170  IF XU = 0 THEN  PRINT "
       FILE NOT FOUND"
3180  FOR XX = 1 TO 1500: NEXT
3190  RETURN
3500  IF  LEN (S$) < 30 THEN S$ =
     S$ +  MID$ ("
                    ",1,30 -  LEN
     (S$)): REM  30 SPACES
3510 XU = 1
3520  FOR XZ = 1 TO 29
3530  IF  PEEK (BUF + XY + XZ + 2
     ) <  >  ASC ( MID$ (S$,XZ,1)
     ) + 128 THEN XU = 0:XZ = 30
3540  NEXT
3550  RETURN
4000  REM  REMOVE DOS
4010  HOME : VTAB 7
4020  PRINT " IF DOS IS REMOVED Y
     OU CAN NOT BOOT THE": PRINT
```

```
4030  PRINT " DISK.  DO YOU WANT
    TO CONTINUE? (Y/N)"
4040  FOR XX = 1 TO 2 STEP 0
4050  VTAB 9: HTAB 39: INPUT "";S
    $
4060  IF S$ = "N" OR S$ = "Y" THEN
    XX = 2
4070  NEXT
4080  IF S$ = "N" THEN  RETURN
4090  POKE ITRK,17: POKE ISEC,0
4100  POKE ICMD,RD: CALL RWTS: REM
     READ THE VTOC
4110  IF  PEEK (IERR) <  > 0 THEN
    GOSUB 11000: RETURN
4120  POKE BUF + 56,0: POKE BUF +
    57,0: REM   TRK 0 NOT AVAILA
    BLE
4130  FOR XX = 60 TO 64 STEP 4: POKE
    BUF + XX,255: POKE BUFF + X +
    1,255: NEXT
4140  POKE ICMD,WR: CALL RWTS
4150  IF  PEEK (IERR) <  > 0 THEN
    GOSUB 11000: RETURN
4160  FOR XX = 1 TO 118: POKE BUF
    + XX - 1,NDB(XX): NEXT
4170  FOR XX = 118 TO 255: POKE B
    UF + XX,0: NEXT
4180  POKE ITRK,0: POKE ISEC,0
4190  CALL RWTS: REM   WRITE BOOT
    SECTOR
4200  IF  PEEK (IERR) <  > 0 THEN
    GOSUB 11000
4210  RETURN
5000  HOME : VTAB 6
5010  HTAB 10: PRINT "CURRENT SLO
    T:"; PEEK (IBSL) / 16: PRINT
5020  HTAB 9: PRINT "CURRENT DRIV
    E:"; PEEK (IDRV): PRINT
5030  FOR XX = 1 TO 2 STEP 0: VTAB
    12: HTAB 14: INPUT "NEW SLOT
    :";S$:S$ =  LEFT$ (S$,1): IF
    S$ > "0" AND S$ < "8" THEN X
    X = 2
5040  NEXT
5050  POKE IBSL, VAL (S$) * 16
5060  FOR XX = 1 TO 2 STEP 0: VTAB
    14: HTAB 13: INPUT "NEW DRIV
    E:";S$:S$ =  LEFT$ (S$,1): IF
    S$ > "0" AND S$ < "3" THEN X
    X = 2
5070  NEXT
5080  POKE IDRV, VAL (S$)
5090  RETURN
6000 I = 2: RETURN
10000  REM  ***** SETUP *****
10010  REM  INSTALL INTERFACES
10020  FOR XX = 770 TO 833: READ
    XY: POKE XX,XY: NEXT
10030  CALL 770: REM   GET IOB AD
    DR
10040 IOB =  PEEK (6) + ( PEEK (7
    ) * 256)
10050 IBSL = IOB + 1
10060 IDRV = IOB + 2
10070 IVOL = IOB + 3
10080 ITRK = IOB + 4
10090 ISEC = IOB + 5
10100 ICMD = IOB + 12
10110 IERR = IOB + 13
10120 IAVOL = IOB + 14
10130 BUF =  PEEK (IOB + 8) + ( PEEK
    (IOB + 9) * 256)
10140  POKE IVOL,0: REM   CLEAR V
    OLUME ID
10150 RWTS = 780
10160 RD = 1:WR = 2
10170  TEXT : HOME
10180  PRINT "  * * * *    DISK D
    OCTOR    * * * * ": POKE 34,
    2
10190 PRBYTE = 800
10200 ESCF = 64578
10210 BYTE = 8
10220 ZAP = 810:BDISP = 26: REM
    ZAP VARIABLES
10230  POKE 27, PEEK (IOB + 8): POKE
    28, PEEK (IOB + 9)
10240  DIM NDB(118)
10250  FOR XX = 1 TO 118: READ ND
    B(XX): NEXT
10260  DIM ZTBL(19)
10270  FOR I = 0 TO 19: READ ZTBL
    (I): NEXT
10280  RETURN
10290  REM  RWTS INTERFACE
10300  DATA 32,227,3,133,7,132,6,
    96,234,234
10310  DATA 165,7,164,6,32,217,3,
    176,6
10320  DATA  169,0,160,13,145,6,9
    6
10330  DATA  234,234,234,234
10340  REM  MONITOR PRBYTE INTERF
    ACE
10350  DATA  165,8,32,218,253,96
10360  REM  NIBBLE ZAP ROUTINE
10370  DATA  234,234,234,234,162,
    240,165,8,16
10380  DATA  8,10,10,10,10,133,8,
    162,15,138
10390  DATA 164,26,49,27,5,8,145,
    27,96
10400  REM  BOOT ROUTINE WHEN DOS
     REMOVED
10410  DATA 1,157,136,192,142,244
    ,3,138,74
10420  DATA  74,74,74,9,192,141,4
    5,8,165
10430  DATA 0,141,44,8,32,96,251,
    162,0
10440  DATA  189,46,8,240,6,32,24
    0,253,232
10450  DATA 208,245,32,111,253,10
     8,44,8,0
10460  DATA 0,141,141,160,160,160
    ,160,160,160
10470  DATA 160,160,160,206,207,1
    60,196,207,211
10480  DATA 160,207,206,160,212,2
    00,201,211,160
10490  DATA 196,201,211,203,141,1
    41,160,160,201
10500  DATA 206,211,197,210,212,1
    60,196,201,211
10510  DATA 203,160,215,201,212,2
    00,160,196,207
10520  DATA 211,160,193,206,196,1
    60,208,210,197
10530  DATA 211,211,160,210,197,2
    12,213,210,206
10540  DATA   0
10550  REM
10560  REM  TAB TABLE FOR ZAP
10570  REM
10580  DATA  30,30,31,31,0,32,32,
    33,33,0,34,34,35,35,0,36,36,
    37,37,0
10970  REM
10980  REM  ---- DISK ERROR HANDL
    ER ----
10990  REM
11000  HOME :XX =  PEEK (IERR)
11010  IF XX = 16 THEN S$ = " DIS
    K IS WRITE PROTECTED "
11020  IF XX = 32 THEN S$ = "
     VOLUME MISMATCH     "
11030  IF XX = 64 THEN S$ = "
       I/O ERROR        "
11040  IF XX = 128 THEN S$ = "
          READ ERROR        "
11050  INVERSE : VTAB 7: HTAB 8: PRINT
     SPC( 25): VTAB 8: HTAB 8: PRINT
    S$: VTAB 9: HTAB 8: PRINT  SPC(
    25)
11060  FOR XX = 1 TO 2500: NEXT
11070  NORMAL
11080  RETURN
```

Raise your Apple's IQ Twelve Times a year!

Subscribe Now and Save $12.00 off the Cover Price

A one-year subscription to Nibble brings you twelve issues packed with programs and comprehensive articles to help you get the most out of your Apple.

You'll get over $500 worth of programs for home/personal finance, business, programming utilities, education, and entertainment, all with complete instructions. We even show you how they work, how you can customize them, and how you can use similar techniques to write your own programs! And Nibble programs are available on disk or in Cauzin data strip format, at prices you can afford.

Whether you're a beginner or expert, you'll enjoy regular features such as Graphics Workshop, Assembly Language for Beginners, Educational Comer, and Tips and Techniques. Nibble also features insightful, comparative product reviews, tutorials and tips on programming, techniques for using popular commercial software more efficiently, and information on the business aspects of professional and home computing.

So fill out the coupon today and start your monthly subscription to Nibble, the only Apple magazine that gives you something to *do* with your computer.



Clip the coupon below and mail it to:
Nibble, 45 Winthrop St., Concord, MA 01742

( ) Yes! I'll try Nibble! Enclosed is:

( ) $26.95 for 12 issues ( ) $49.95 for 24 issues
( ) $69.95 for 36 issues

NAME ADDRESS

CITY STATE ZIP CODE

( ) Check/money order enclosed

Charge my ( ) VISA ( ) MasterCard ( ) Bill me **(U.S. only)**

Credit Card Number Expiration Date

Signature Telephone Number

Nibble Magazine Foreign Subscription Rates: Canada Surface $34.95; Canada Air Mail $59.95; Outside U.S. and Canada Surface $39.95; Outside U.S. and Canada Air Mail $89.95. All payments must be in U.S. funds drawn on a U.S. bank.

C2SW-71

LIMITED WARRANTY

Cauzin StripWare™

835 South Main Street
Waterbury, Connecticut 06706

CAUZIN SYSTEMS, INC. (CAUZIN) warrants this package against defects in materials and workmanship for a period of 90 days from the date of original purchase.

If you discover a defect, CAUZIN will, at its option, either repair, replace, or refund the purchase price of this product at no charge to you, provided it is returned during the warranty period indicated above or any extended period(s) paid for by the user. Transportation charges should be prepaid, to the authorized dealer from whom you purchased it or to any other authorized CAUZIN dealer in the country.

You may obtain additional information from CAUZIN directly at the address printed on this certificate, or by calling (203) 573-0150. Please attach your name, address, telephone number, a description of the problem, and a copy of a bill of sale as proof of date of original purchase, to each product returned for warranty service.

This warranty applies only to products manufactured by or for CAUZIN, which can be identified by the CAUZIN trademark, trade name, or logo affixed to the product. CAUZIN does not warrant any product(s) that are not CAUZIN products. This warranty does not apply if the product has been damaged by accident, abuse, misuse or misapplication, has been modified without the written permission of CAUZIN, or if any CAUZIN serial number has been removed, altered, or defaced.

ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, ARE LIMITED IN DURATION TO 90 DAYS FROM THE ORIGINAL DATE OF PURCHASE. THERE ARE NO WARRANTIES WHICH EXTEND BEYOND THE DESCRIPTION ON THE FACE HEREOF.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESSED OR IMPLIED. NO CAUZIN DEALER, AGENT, OR EMPLOYEE IS AUTHORIZED TO MAKE ANY MODIFICATION, OR ADDITION TO THIS WARRANTY.

CAUZIN IS NOT RESPONSIBLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY BREACH OF WARRANTY, OR UNDER ANY OTHER LEGAL THEORY, INCLUDING LOST PROFITS, DOWNTIME, GOODWILL, DAMAGE TO, OR REPLACEMENT OF EQUIPMENT AND PROPERTY, AND ANY COST OF RECOVERING, REPROGRAMMING OR REPRODUCING ANY PROGRAM OR DATA STORED IN OR USED WITH CAUZIN PRODUCTS.

Somes states do not allow the exclusion or limitation of incidental or consequential damages or limitations on how long an implied warranty lasts, so the above limitations or exclusion may not apply to you.

This warranty shall not be applicable to the extent that any provision of this warranty is prohibited by an federal, state or municipal law which cannot be preempted. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.



OTHER TITLES AVAILABLE IN STRIPWARE tm



FOR YOUR APPLE

Art and Graphics on the Apple II/IIe
BASIC Apple IIc
Home Education, Vol. 1
Arcade Games, Vol. 1
Disk Doctor
Computer Puzzles
Balance Sheet
Essential Programs for Small Business Planning





FOR IBM

PC Graphics
Computer Puzzles
Balance Sheet
Classic Games
Eliza
Home Applications
Personal Calendar
Celestia





Softstrip
COMPUTER READABLE PRINT

- Display and edit information directly from sectors of your disk
- Modify disk directories
- Read and write control characters
- Examine hidden file names
- Diagnose disk illnesses and cure them

CAUZIN SYSTEMS, INC.
835 South Main Street
Waterbury, Connecticut 06706
(203) 573-0150

850.2002